
 GOBIERNO DEL ESTADO DE MÉXICO	MANUAL DE PRÁCTICAS FO-TESJI-11100-12	 TESJI TECNOLÓGICO DE ESTUDIOS SUPERIORES JILOTEPEC
---	---	---

NOMBRE DE LA PRÁCTICA	APUNTADORES			No.	6
ASIGNATURA:	METODOS NUMERICOS	CARRERA:	INGENIERÍA EN SISTEMAS COMPUTACIONALES	PLAN:	ISIC-2010-224

NOMBRE: YOLOTZIN DOMINGUEZ SANTOS
GRUPO: 3041

I. COMPETENCIA(S) ESPECÍFICA(S):

II. MATERIAL EMPLEADO:

- Dev-C++

III. DESARROLLO DE LA PRÁCTICA:

APUNTADORES

Un puntero es un objeto que apunta a otro objeto.
Es decir, una variable cuyo valor es la dirección de memoria de otra variable.

En C no se debe indicar numéricamente la dirección de la memoria, si no que se usa una etiqueta que conocemos como **variable**.

Las direcciones de memoria dependen de la Arquitectura del ordenador y de la gestión que el Sistema operativo haga de ella.

¿Cómo se declaran los apuntadores?

Para declarar un apuntador se especifica el tipo de dato al que apunta, el operador '*', y el nombre del apuntador.

Un puntero tiene su propia dirección de memoria.

La sintaxis es la siguiente:

```
<tipo de dato apuntador> * <identificador del apuntador>
int* punt;
char* car;
float* num;
```

LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):	<u>LABORATORIO DE COMPUTO</u>	DURACIÓN DE LA PRÁCTICA (HRS):	<u>12 horas</u>
Elaboró	Versión		
Representante de la Dirección	1		
Autorizó	Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec	7 de Febrero de 2017		

Tipos de apuntadores

Hay tantos tipos de apuntadores como tipos de datos.

Se puede también declarar apuntadores a estructuras más complejas.

Funciones

Struct

Ficheros

Se pueden declarar punteros vacíos o nulos.

¿Qué es la referenciación?

La referenciación es obtener la dirección de una variable.

Se hace a través del operador '&', aplicado a la variable a la cual se desea saber su dirección

Fragmento de código- referenciación

Ejemplo 1:

```
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5
6
7     int numero =1;
8     int x=17,y;
9     int * p;
10    p= &x;
11    printf("El valor de x es %d \n", *p);
12    y=*p+3;
13    printf("El valor de y es %d \n",y);
14
15    system ("Pause");
16    return 0;
17 }
```

```
C:\Users\VOLOTIIN\Documents\Zamora Dev
El valor de x es 17
El valor de y es 20
Presione una tecla para continuar . . .
```

Dentro del método main :

Guardaremos dos variables de tipo int x=17, y n tendrá valor

Y otra variable *p de indirección

En p guardaremos &x es decir de dirección

Mandaremos el mensaje y veremos cual es valor de x con *P n afecta porque es como si igualáramos esas variable va guardar el valor original de x=17

Y para y será lo mismo ya que *p guarda el valor de la x original entonces solo sumamos 3 =20

LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):		LABORATORIO DE COMPUTO	DURACIÓN DE LA PRÁCTICA (HRS):	12 horas
Elaboró		Versión		
Representante de la Dirección		1		
Autorizó		Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec		7 de Febrero de 2017		

Ejemplo 2:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5     int *p, y;
6
7     void func() {
8
9         int x= 40;
10        p=&x;
11        y=*p;
12        *p=23;
13    }
14
15    int main() {
16        func();
17        y=*p;
18        *p=25;
19        printf("El valor de y es %d \n El valor de *p es %d \nEl valor e p es %p\n", y, *p, p);
20
21        system ("Pause");
22        return 0;
23    }
24

```

Dos variables de tipo entero *p, y
Dentro del método func ya que no
tiene retorno;
Pondremos las operaciones en el x=40,
Mientras p=tendrá el valor de x,
mientras que y=*p obtendrá el el valor
de *p
Dentro del método main llamaremos al
método func y como ahí y = 23 aquí
*p valdrá 25, y p = 0028FEF4

```

El valor de y es 23
El valor de *p es 25
El valor e p es 0028FEF4
Presione una tecla para continuar . . . _

```

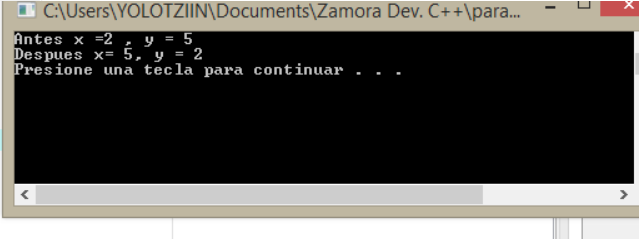
LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):		LABORATORIO DE COMPUTO	DURACIÓN DE LA PRÁCTICA (HRS):	12 horas
Elaboró		Versión		
Representante de la Dirección		1		
Autorizó		Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec		7 de Febrero de 2017		

Ejemplo 3:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void) {
5
6     int x = 2;
7     int y = 5;
8
9     printf("Antes x = %d , y = %d \n", x, y);
10    intercambio (&x, &y);
11    printf("Despues x= %d, y = %d\n", x, y);
12    system ("Pause");
13 }
14
15 void intercambio (int *a, int *b) {
16     int temp;
17     temp = *b;
18     *b = *a;
19     *a = temp;
20
21 }
22

```



Dentro del método void mandaremos sin retorno, dentro de seste habrá dos variables x = 2 y y = 5

Imprimiremos su valor de x y y, después aremos el intercambio utilizando el método sin retorno de intercambio,

Que tiene variables de tipo entero: Donde la variable temp = guardara el valor de *b = *a y *a = temp Ya que esto seria igual (&x, &y) que (int *a , *b) por eso al hacer la igualación quedaría de la siguiente manera:

La función sizeof()

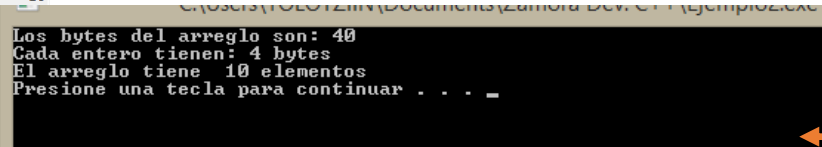
✉ Devuelve el tamaño en bytes que ocupa un tipo o variable en memoria.

Ejemplo 4:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main() {
5
6     int array[10]= {1,2,3,4,5,6,7,8,9,0};
7     int len=sizeof(array)/sizeof(int);
8     printf("Los bytes del arreglo son: %d\n", sizeof(array));
9     printf("Cada entero tienen: %d bytes\n", sizeof(int));
10    printf("El arreglo tiene %d elementos\n", len);
11
12    system ("Pause");
13    return 0;
14 }
15

```



En el método main, tenemos un array con un arreglo de 10 posiciones que estas van del 1 al 0.

Con len lo utilizamos para medir entonces con sizeof veremos cuantos bytes tiene array y cuantos tiene int.

En primer lugar imprimiríamos los bytes del arreglo, después los bytes de el entero y al final cuantos elementos tiene el arreglo.

LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):		LABORATORIO DE COMPUTO	PRÁCTICA (HRS):	12 horas
Elaboró		Versión		
Representante de la Dirección		1		
Autorizó		Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec		7 de Febrero de 2017		

Asignación dinámica de memoria

```

while.c | referencion.c | Apuntadores1.c | parametrosPorReferencia.c | Ejemplo2.c | AsignacionDinamica.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int main(void){
6     int i,n;
7     char * buffer;
8     printf (" Teclea la longitud de la cadena? ");
9     scanf ("%d", &i);
10    buffer = (char*) malloc (i+1);
11    if (buffer==NULL) exit (1);
12    for (n=0; n<i; n++)
13        buffer[n]=rand()%26+'a';
14    buffer[i]='\0';
15    printf ("Random string: %s\n",buffer);
16    free (buffer);
17    system("Pause");
18 }
19
C:\Users\YOLOTZIIN\Documents\Zamora Dev. C++\AsignacionDinamica.exe
Teclea la longitud de la cadena? 16
Random string: phqghumeaylnlfdx
Presione una tecla para continuar . . . _

```

Dentro del método main tendremos las variables de tipo entero i,n y un char buffer, Pediremos el dato desde teclado y lo guardaremos en i , después con buffer liberaremos lo que tiene char y el retorna NULL en caso de no haber conseguido suficiente memoria. Pero en este caso no retorno ningún null porque n ocupamos mucha memoria,

Con free =permite liberar la memoria

Reservada a través de un apuntador. Cuando imprimimos obtenemos en letras la cantidad de longitud de cadena.

LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):		LABORATORIO DE COMPUTO	DURACIÓN DE LA PRÁCTICA (HRS):	12 horas
Elaboró		Versión		
Representante de la Dirección		1		
Autorizó		Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec		7 de Febrero de 2017		

Ejercicio:

- *Crea un arreglo entero de tamaño x, en donde x es ingresado por teclado.
- *Llena todos los elementos del arreglo con datos ingresados por el usuario.
- *Muestra los valores

```

Ejercicio01...cpp  referencion.c  ejer01.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void){
5
6     int i,n;
7
8     printf("Ingrese cantidad de arreglo: ");
9     scanf("%d",&n);
10    int x[n];
11    for(i=0; i<n ; i++){
12        printf("Ingresa el %d \n", (i+1));
13        scanf("%d",&x[i]);
14    }
15    for(i=0; i<n ; i++){
16        printf("\nEl numero es: %d\n", (i+1),x[i]);
17    }
18    system ("Pause");
19    return 0;
20 }
21

```

```

C:\Users\YOLOTZIIN\Documents\Zamora Dev. C++\ejer01.exe
Ingrese cantidad de arreglo: 7
7
Ingresa el 1
7
Ingresa el 2
6
Ingresa el 3
5
Ingresa el 4
4
Ingresa el 5
3
Ingresa el 6
2
Ingresa el 7
1
El numero es: 1
El numero es: 2
El numero es: 3
El numero es: 4
El numero es: 5
El numero es: 6
El numero es: 7
Presione una tecla para continuar . . .

```

LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):		LABORATORIO DE COMPUTO	DURACIÓN DE LA PRÁCTICA (HRS):	12 horas
Elaboró		Versión		
Representante de la Dirección		1		
Autorizó		Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec		7 de Febrero de 2017		

Direcciones del arreglo

```
while.c | referencion.c | Apuntadores1.c | parametrosPorReferencia.c | Ejemplo2.c | AsignacionDinamica.c | Direcciones.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4     int i[10], x;
5     float f[10];
6
7     int main(void){
8         printf (" \t\tEntero\t\t\tFlotante\n\n");
9         for(x=0; x<10 ; x++){
10             printf (" El elemnto %d:\t%d\t\t\t\t%f\n",x,&i[x], &f[x])
11         }
12
13         system ("Pause");
14
15     }
16
```

```
C:\Users\YOLOTZINI\Documents\Zamora Dev. C++\Direcciones.e
      Entero      Flotante
El elemnto 0: 4214912      4214976
El elemnto 1: 4214916      4214980
El elemnto 2: 4214920      4214984
El elemnto 3: 4214924      4214988
El elemnto 4: 4214928      4214992
El elemnto 5: 4214932      4214996
El elemnto 6: 4214936      4215000
El elemnto 7: 4214940      4215004
El elemnto 8: 4214944      4215008
El elemnto 9: 4214948      4215012
Presione una tecla para continuar . . . _
```

Fuera del método main tendremos las variables de tipo entero i y fe con un arreglo de 10 posiciones.

Dentro del método main, con un metodo sin retorno , imprimiremos Entero y Flotante t dar con espacios para poderlos alinear.

Luego con el método for evaluaremos las condiciones, y primero es la misma cantidad de datos que pueden tener ese espacio luego te muestran que tanto cantidad son en entero y en flotante.

LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):		LABORATORIO DE COMPUTO	DURACIÓN DE LA PRÁCTICA (HRS):	12 horas
Elaboró		Versión		
Representante de la Dirección		1		
Autorizó		Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec		7 de Febrero de 2017		

Aritmética de operadores

Ejercicio

*Crea un arreglo entero de tamaño x, en donde x es ingresado por teclado.

*Llena todos los elementos del arreglo con datos ingresados por el usuario usando apuntadores.

```

Ejercicio01...cpp | referencion.c | ejer01.c | [*] Ej02.c
1 #include <stdio.h>
2 #include <stdlib.h>
3
4 int main(void){
5     int i,n;
6
7     printf("Ingrese cantidad de arreglo: ");
8     scanf("%d",&n);
9     int x[n];
10    for(i=0; i<n ; i++){
11        printf("Ingresa el %d = ",(i+1));
12        scanf("%d",&x[i]);
13    }
14    for(i=0; i<n ; i++){
15        printf("\nEl %d numero es = %d\n",(i+1),x[i]);
16    }
17    system("Pause");
18    return 0;
19 }
20
C:\Users\YOLOTZIIN\Documents\Zamora Dev. C++\Ejr01
Ingrese cantidad de arreglo: 3
3
Ingresa el 1 = 34
Ingresa el 2 = 56
Ingresa el 3 = 67
El 1 numero es = 3
El 2 numero es = 34
El 3 numero es = 56
Presione una tecla para continuar . . . _

```

LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):		LABORATORIO DE COMPUTO	DURACIÓN DE LA PRÁCTICA (HRS):	12 horas
Elaboró		Versión		
Representante de la Dirección		1		
Autorizó		Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec		7 de Febrero de 2017		

Ejemplo:

```

1 #include <stdio.h>
2 #include <stdlib.h>
3
4
5 int main(void){
6     int i,n;
7     int * buffer, * p_buffer;
8
9
10    printf (" Teclea la longitud del arreglo ");
11    scanf ("%d", &n);
12
13    buffer = (int*) malloc((n) * sizeof(int) );
14    if(buffer==NULL) exit (i);
15
16    p_buffer = buffer;
17    for(i=0; i<n; i++){
18        printf (" Ingrese el valor %d \n",i);
19        scanf("%d",p_buffer++);
20    }
21
22    p_buffer = buffer;
23    printf("\nLos valores son \n");
24    for(n=0; n<i; n++){
25        printf("arreglos [%d] = %d \n", n, *p_buffer++);
26    }
27
28    free (buffer);
29    system ("Pause");
30 }
31

```

Dentro del método main que no tendrá retorno, tendremos las variables de tipo entero i,n y * buffer y *p_buffer.

Pediremos el dato desde teclado y lo guardaremos en i , después con buffer liberaremos lo que tiene char y el retorna NULL en caso de no haber conseguido suficiente memoria. Pero en este caso no retorno ningún null porque n ocupamos mucha memoria,

Y hasta que terminara evaluando todas las condiciones que tienen que cumplir para imprimir lo que falta del resultado ya que debe de ir de posición en posición del arreglo.

Cuando imprimimos obtenemos un ciclo dependiendo de la cantidad de cadena y nos pedirá los valores y después los imprimirá en orden que los ingresamos.

*Crea un arreglo de tipo char de tamaño x, en donde x es ingresado por teclado.

*Llena elemento por elemento del arreglo con letras ingresados por el usuario.

* Muestra el arreglo impreso en forma inversa.

*Todo debe ser manejado con apuntadores.

LUGAR DE REALIZACIÓN DE LA PRÁCTICA (LABORATORIO/TALLER/AULA):	LABORATORIO DE COMPUTO	DURACIÓN DE LA PRÁCTICA (HRS):	12 horas
Elaboró	Versión 1		
Representante de la Dirección			
Autorizó	Fecha de revisión		
Director General del Tecnológico de Estudios Superiores de Jilotepec	7 de Febrero de 2017		